

GrowCLIP: Data-aware Automatic Model Growing for Large-scale Contrastive Language-Image Pre-training

Xinchi Deng¹ Han Shi² Runhui Huang¹ Changlin Li³ Hang Xu²
Jianhua Han² James Kwok⁴ Shen Zhao^{1*} Wei Zhang² Xiaodan Liang¹

¹Sun Yat-sen University ²Huawei Noah’s Ark Lab

³University of Technology Sydney ⁴The Hong Kong University of Science and Technology

dengxch5@mail2.sysu.edu.cn

Abstract

Cross-modal pre-training has shown impressive performance on a wide range of downstream tasks, benefiting from massive image-text pairs collected from the Internet. In practice, online data are growing constantly, highlighting the importance of the ability of pre-trained model to learn from data that is continuously growing. Existing works on cross-modal pre-training mainly focus on training a network with fixed architecture. However, it is impractical to limit the model capacity when considering the continuously growing nature of pre-training data in real-world applications. On the other hand, it is important to utilize the knowledge in the current model to obtain efficient training and better performance. To address the above issues, in this paper, we propose GrowCLIP, a data-driven automatic model growing algorithm for contrastive language-image pre-training with continuous image-text pairs as input. Specially, we adopt a dynamic growth space and seek out the optimal architecture at each growth step to adapt to online learning scenarios. And the shared encoder is proposed in our growth space to enhance the degree of cross-modal fusion. Besides, we explore the effect of growth in different dimensions, which could provide future references for the design of cross-modal model architecture. Finally, we employ parameter inheriting with momentum (PIM) to maintain the previous knowledge and address the issue of the local minimum dilemma. Compared with the existing methods, GrowCLIP improves 2.3% average top-1 accuracy on zero-shot image classification of 9 downstream tasks. As for zero-shot image retrieval, GrowCLIP can improve 1.2% for top-1 image-to-text recall on Flickr30K dataset.

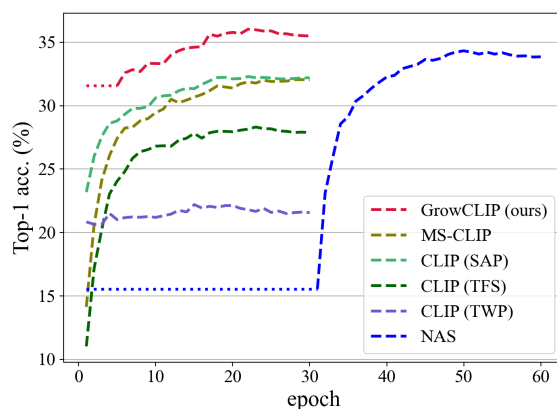


Figure 1. Top-1 accuracy (%) of zero-shot image classification on ImageNet of GrowCLIP and baselines during training at step 4, where the horizontal dotted lines mean the process of supernet training. Our GrowCLIP has the best performance and is more efficient compared with other baselines.

1. Introduction

Recently, large-scale pre-trained models illustrate the potential performance among different fields including computer vision (CV) [23, 24, 11] and natural language processing (NLP) [15, 43, 32]. Cross-modal models like CLIP [51], ALIGN [27], FILIP [64], BLIP [35] also demonstrate remarkable success across various vision-language downstream tasks. Note that these models are usually built in dual-stream architectures, which consist of the image and text encoders to extract the feature of image and text inputs, respectively. The alignment between these two features is then performed under the contrastive objective to learn the alignment between different modals.

However, training these cross-modal models requires a large amount of image-text pairs collected from the Internet, e.g., 400M image-text pairs for CLIP [51] and 300M for FILIP [64]. Most existing cross-modal methods [51, 27, 64, 38, 35, 68, 61, 1] directly train the final large-

*Corresponding author.

CC12M	10%	20%	50%	100%
CLIP-0.5-ViT-B/16	6.6	11.1	19.0	25.7
CLIP-ViT-B/16	6.5	11.6	20.9	28.3

Table 1. Result of different model sizes with different data sizes: Top-1 accuracy (%) of zero-shot image classification on ImageNet. The depth and the width of 0.5-ViT-B/16 are both one half of ViT-B/16. The relative performance ranking is dependent on the size of the training data.

scale model using the completed image-text pair datasets collected at some point in time. However, it is impractical to regard cross-modal pre-training as “disposable” training without considering the continuously growing nature of pre-training data or domain in real-world applications. For instance, mountains of new knowledge is generated on the Internet every day, which can be used to further improve the performance of existing pre-trained cross-modal models. Therefore, in this paper, we consider the training of large-scale cross-modal pre-trained models as an online learning problem [17, 54, 55] with continuous image-text pairs as input. Different from the standard continual learning setting [57, 63, 21, 30, 2, 7], the previous image-text pairs can also be achieved in our setting since we assume the size of training data is gradually increased over time.

One tough problem in the online learning scenario is that the model capacity should be related to the size of training data [16, 69, 28, 25]. For example, it’s observed that large ViT models perform worse than ResNets when pre-trained on small datasets, while the result is opposite when they are both pre-trained on larger datasets [16]. To verify the relationship between model capability and data size, we split Conceptual 12M (CC12M) dataset [9] and test the performance of ViT-B/16 with different scales [51]. As shown in Table 1, the relative performance ranking is dependent on the size of the training data. Given a smaller dataset (e.g., 10% CC12M), the performance of 0.5-ViT-B/16 is comparable with ViT-B/16. In contrast, when data is sufficient (e.g., 100% CC12M), the performance of ViT-B/16 is better than the smaller model. Thus, the unchanged model is not practical for this real-time learning setting and it’s still an open issue on how to modify and train our model with the growing data. Besides, how to efficiently make use of knowledge of previous model when new data is coming remains an open problem. One direct solution is to fine-tune the model with the updated training dataset. However, training with previous pre-trained parameters of the same model will deteriorate the performance [4] due to the parameter inheriting issue. As shown in Figure 1, the CLIP training with pre-training (TWP) have worse performance than the one training from scratch (TFS).

To address the above issues, we propose a data-aware automatic model growing method (denoted as GrowCLIP) for large-scale contrastive language-image pre-training, which

performs a model growth process considering the gradually increased pre-training data. Specially, when training data grows dynamically, we adopt different cross-modal network architectures via the customized neural architecture search (NAS) to make the network pre-training more efficient. Different from traditional NAS approaches [70, 52, 62], we introduce a cross-modal customized NAS by defining a dynamic search space named growth space, which is enlarged when more data comes, and proposing a shared encoder search space to enhance the degree of cross-modal fusion. To utilize the architecture at the previous step more efficiently, the parameters of the new architecture are also inherited from the old one with momentum to maintain the previous knowledge and address the issue of local minimum dilemma. Finally, growth architecture selection procedure is performed to select the optimal model architecture at each step, considering the performance and model size.

Experiments are conducted by averagely dividing the Conceptual 12M (CC12M) into 4 growth steps under the online learning setting. As depicted in Table 1, compared with the existing methods, our GrowCLIP has the best performance and is more efficient. Specially, experimental results show that our GrowCLIP can improve up to 2.3% average top-1 accuracy on zero-shot image classification of 9 downstream tasks compared with the existing methods. As for zero-shot image-text retrieval, GrowCLIP has a 1.2% improvement for top-1 image-to-text recall on Flickr30K [49] dataset and 0.8% on MSCOCO [39].

To summarize, the contributions of this paper are listed as follows: (i) To adapt to the growing data scenario, we propose a data-aware automatic model growing method, named GrowCLIP. (ii) We provide some insights for the design of cross-modal model architecture. (iii) The extensive experiment results illustrate the effectiveness of our approach on zero-shot classification and retrieval tasks.

2. Related Work

Vision-Language Pre-training Models Inspired by the success of pre-training in computer vision (CV) and natural language processing (NLP), a boosting number of research works in the domain of vision-language pre-training (VLP) has recently surged to pursue a unified multi-modality representation. Vision-language pre-training trains the model on large-scale image-text pairs to improve performance of downstream vision and language tasks including image classification [14, 34, 31], retrieval [49, 39], Image Captioning (IC) [42, 13, 26, 25], Visual Question and Answer (VQA) [3, 45, 67] and so on. The VLP model can be categorized into two categories: (i) The dual-stream model, e.g., CLIP [51], ALIGN [27], FILIP [64], processes visual and textual tokens separately with two parallel streams to acquire the representation and then fuse them through interactive module. (ii) The single-

stream model, e.g., Oscar [37], ViLT [29], VisualBERT [36], directly combines the extracted visual and textual tokens and feeds them into the transformer-based model. The public pre-training datasets of VLP model include available datasets like YFCC100M [59] and CC12M [9]. CLIP [51], ALIGN [27], FILIP [64] and LEMON [25] adopt large-scale image-text pairs crawled from the web to show more powerful models. Different from these works that treat this process as offline learning, we investigate the online learning case in vision-language pre-training.

Online Learning Online learning is a problem setting of machine learning in which data becomes available in a sequential order [17, 54, 55]. Different from continual learning [21, 30, 2, 7], online learning can reach previous data all the time. As opposed to batch learning techniques which generate the best model by learning on the entire training dataset at once, online learning is used to update our predictor for future data at each growth step. Follow the leader [56] is the simplest learning rule which assumes the hypothesis that the leader has the least loss overall past rounds. Follow the regularised leader [50] is a modification variant and obtain better regret bounds by learning a regularisation function. Online convex optimization [22] is a general algorithm framework for decision making which targets convex optimization. To adapt online learning scenario, we propose a data-aware automatic model growing algorithm that grows the model with increasing training data.

Neural Architecture Search Neural Architecture Search (NAS) aims to automatically design the state-of-the-art neural networks. Early works [70, 53, 58] are based on reinforcement learning and evolutionary algorithms. In the later work, various techniques are used to reduce the computational complexity. Gradient-based methods [41, 40] regard the network architecture as a set of parameters and adopt back-propagation algorithm to optimize them. One-shot NAS methods [48, 10, 20] train a supernet to avoid training each subnet from scratch. Unfortunately, one-shot NAS methods have to spend lots of resources to train the supernet. In contrast, the method we propose inherits the parameters of the old model with momentum, which only takes few epochs to train the supernet. Furthermore, GrowCLIP can modify the neural architecture to adapt to the size of the training data.

3. Methodology

In this section, we describe the online learning problem setting and its challenges for cross-modal pre-training (Section 3.1). To adapt the growth data, we propose a dynamic growth space, including a proposed shared encoder (Section 3.2). To rise to the challenge of the local minimum dilemma, we propose parameter inheriting with momentum (Section 3.3). The whole pipeline of our method is illus-

trated in Section 3.4.

3.1. Problem Setting and Challenges

Problem Setting The target of online learning is obtaining the optimal model at each growth step of the data. Let $\mathcal{D}^{t-1} = \{\mathbf{x}_i^I, \mathbf{x}_i^T\}_{i=1}^k$ be the training data at growth step $t-1$, where \mathbf{x}^I and \mathbf{x}^T are image and text samples and k is the training data size. At the next growth step t , the enlarged training dataset \mathcal{D}^t satisfies $\mathcal{D}^{t-1} \subseteq \mathcal{D}^t$ due to the incoming data. Given an optimal model from growth step $t-1$, our aim is to achieve the optimal model (with optimal architecture ψ^* and parameters ω^*) based on the enlarged dataset \mathcal{D}^t :

$$\{\psi^*, \omega^*\} = \arg \min_{\psi, \omega} \mathcal{L}(\psi, \omega; \mathcal{D}^t). \quad (1)$$

Challenge of Data Growth As discussed in Table 1, the size of the training dataset is related to the optimal architecture selection. Given a smaller dataset, the small-scale architecture has comparable performance to large-scale architecture but fewer parameters are deployed. When data grows, the small-scale architecture may limit the final performance due to the limited parameters. However, few works notice the effect of data size in online learning settings.

Challenge of Parameter Inheriting As investigated, the performance of model trained from scratch is much better than trained with pre-training. This is probably caused by the influence of the inheriting parameters trained with the previous smaller dataset, which makes it difficult for the model to escape from the local minima in the grown dataset. However, training from scratch is costly due to multiple growth steps in online learning. The balance between training from scratch and training with pre-training is worthy of consideration.

To solve the above problems, we propose an automatic model growing method - GrowCLIP (shown in Figure 2), which can modify the neural architecture at different growth steps in online learning case. To adapt to the growing data scenario, we propose a growth space for each growth step, where the scale of architecture candidates among this space is larger when data grows. To alleviate the parameter inheriting issue with efficiency, we perform parameter inheriting with momentum such that the trade off between training from scratch and training with pre-training is balanced.

3.2. Growth Space

Model Architecture Following the basic neural architecture in CLIP [51], we adopt a dual-stream model with image encoder $f(\cdot)$ and text encoder $g(\cdot)$. To adapt the cross-modality scenarios, we propose a shared encoder $h(\cdot)$, in which the image and text share the transformers except layernorm layers. The shared encoder not only can reduce

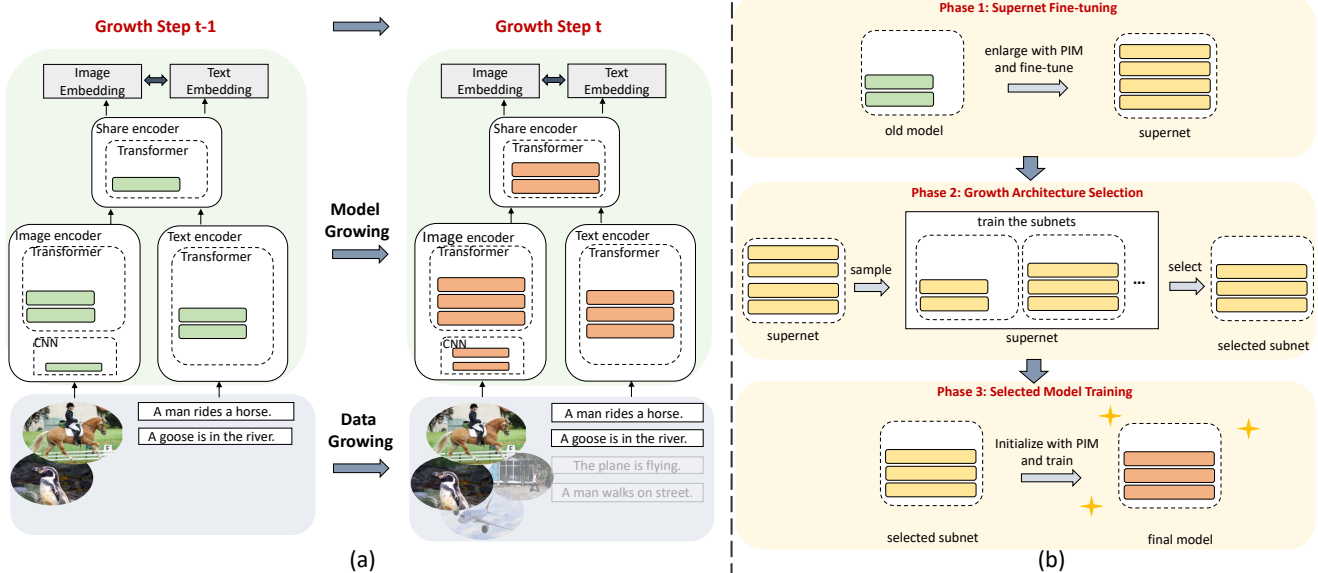


Figure 2. The overview of GrowCLIP. As shown in Figure (a), when the data grows from growth step $t - 1$ to growth step t , our model grows adaptively as well. Figure (b) illustrates the detail of model growing and we simplify the representation of model architecture.

model parameters, but also enhance the degree of cross-modal fusion. Given an image \mathbf{x}_i^I and a text \mathbf{x}_j^T , the model outputs the image representations $I_i = h(f(\mathbf{x}_i^I))$ and text representations $T_j = h(g(\mathbf{x}_j^T))$. The similarity score computation of between image \mathbf{x}_i^I and text \mathbf{x}_j^T is:

$$s_{i,j} = h(f(\mathbf{x}_i^I))^\top h(g(\mathbf{x}_j^T)), \quad (2)$$

where $s_{i,j}$ denotes the similarity of the i -th image to the j -th text.

In each training batch, we sample n image-text pairs $\{\mathbf{x}_i^I, \mathbf{x}_i^T\}_{i=1}^n$ from the dataset. For image \mathbf{x}_i^I in the image-text pair $\langle \mathbf{x}_i^I, \mathbf{x}_i^T \rangle$, \mathbf{x}_i^I is its positive, while the other texts will be used as in-batch negatives. The image-to-text contrastive loss \mathcal{L}_i^I for \mathbf{x}_i^I can be formulated as:

$$\mathcal{L}_i^I(\mathbf{x}_i^I, \{\mathbf{x}_j^T\}_{j=1}^n) = -\frac{1}{n} \log \frac{\exp(s_{i,i})}{\sum_j \exp(s_{i,j})}, \quad (3)$$

Similarly, the text-to-image contrastive loss for \mathbf{x}_i^T is:

$$\mathcal{L}_i^T(\mathbf{x}_i^T, \{\mathbf{x}_j^I\}_{j=1}^n) = -\frac{1}{n} \log \frac{\exp(s_{i,i})}{\sum_j \exp(s_{i,j})}, \quad (4)$$

Totally, the contrastive loss of this mini-batch can be represented by $\mathcal{L} = \frac{1}{2} \sum_{i=1}^n (\mathcal{L}_i^I + \mathcal{L}_i^T)$.

Growth Space of image encoder For the image encoder, we adopt a mixed architecture of convolutional layer and transformer block [60]. There are three variable factors in our growth space design: (i) the number of convolutional layers l^I ; (ii) the number of transformer blocks b^I ; (iii) the number of transformer heads h^I .

Growth Space of text encoder The text encoder part is a transformer-based architecture, whose growth space includes two variable factors: (i) the number of transformer blocks b^T and (ii) the number of transformer heads h^T .

Growth Space of shared encoder The shared encoder is a transformer-based architecture, whose growth space includes the number of transformer blocks b^S .

	Image encoder	Text encoder	Shared encoder
Transformer blocks	$\{b_{t-1}^I, b_{t-1}^I + 4\}$	$\{b_{t-1}^T, b_{t-1}^T + 4\}$	$\{b_{t-1}^S, b_{t-1}^S + 4\}$
Transformer heads	$\{h_{t-1}^I, h_{t-1}^I + 4\}$	$\{h_{t-1}^T, h_{t-1}^T + 4\}$	-
Convolutional layers	$\{l_{t-1}^I, l_{t-1}^I + 2\}$	-	-

Table 2. The growth space Φ of GrowCLIP at the growth step t . $b_{t-1}^I, b_{t-1}^T, b_{t-1}^S, h_{t-1}^I, h_{t-1}^T, h_{t-1}^S$ are the hyperparameters of model architecture at growth step $t - 1$.

The detailed descriptions of growth space for three encoders are shown in Table 2. At each new growth step, the variable factors are enlarged to adapt the growth data. Specifically, the numbers of transformer blocks and transformer heads are both increased by 4, and the number of convolutional layers is increased by 2.

3.3. Parameter inheriting with momentum

Some previous papers utilize the training results of the old model and transfer the knowledge to the new model [18, 19, 24, 4, 33]. Inspired by the above methods, we propose parameter inheriting with momentum (PIM) to avoid local minimum dilemma. The illustration of our parameter inheriting with momentum is shown in Figure 3. To preserve the knowledge from the old model, we extend the size

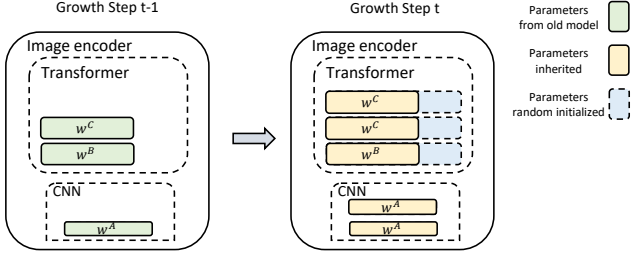


Figure 3. Take image encoder for example of our proposed parameter inheriting with momentum. As for the initialization of the enlarged model, we preserve the existing parameters of the old model for exploitation and combine it with random initialization for exploration.

of the old model to the enlarged model and maintain the parameters in existing parts. As for the extended layers, we copy the parameters from the last layers directly. ω_{old} denotes the parameters of extension of the old model. Besides the above parameter inheriting term, we also add a re-initialization term with randomly initialized parameters ω_{rand} to avoid the parameter inheriting issue. The final initialized supernet is a combination of the above two terms as follows:

$$\omega_{inherit} = \beta \cdot \omega_{old} + \gamma \cdot \omega_{rand}, \quad (5)$$

where β and γ are two hyperparameters related to exploitation and exploration. Additionally, the parameters of new transformer heads are randomly initialized, which makes that different heads in transformer are orthogonal and the new heads learn new features from different perspectives.

3.4. Growth Pipeline

The growth pipeline at growth step t (except the first one) includes three sequential phases as shown in Figure 2 (b).

Phase 1: Supernet Fine-tuning As for a new growth step, a supernet Ψ covering the whole growth space is constructed by enlarging the current model. First, we use the PIM method described in Section 3.3 to initialize the supernet. However, there are some newly added layers in the supernet compared with the old model from the last growth step, which brings a performance gap between the subnets with new layers and others. To make the supernet more robust, we slightly fine-tune the parameters of the full supernet for a few epochs to mitigate the performance gap among subnetworks after PIM initialization.

Phase 2: Growth Architecture Selection Growth Architecture Selection (GAS) is the main phase in our algorithm where we select the optimal architecture in the growth space. The whole GAS process can be divided into two steps: training and searching.

First, to find the optimal architecture ψ^* , we need to fully optimize all the subnet in the supernet, which covers all the candidate growth architecture:

$$\omega_{super}^* = \arg \min_{\omega_{super}} \mathbb{E}_{\psi \in \Psi} \{ \mathcal{L}(\psi, \omega_{super}; \mathcal{D}^t) \}. \quad (6)$$

Inspired by one-shot neural architecture search [8], we optimize ω_{super} in Equation 6 by uniformly sampling subnets ψ from the growth space Φ and update the parameters of the sampled subnets in each iteration. More specifically, at each step, we first randomly sample a subnet ψ from the growth space Ψ , then optimize the parameter of this subnet by minimizing the contrastive loss between image and text. This can be considered as solving Equation 6 via Monte Carlo sampling. By iterative training, these subnets can be fully optimized in a few epochs.

Second, after training the supernet, we search for the optimal subnet in the growth space Φ . Here, we consider both the performance and the number of parameters of the architecture, and select the optimal architecture ψ^* using the composite metric:

$$\psi^* = \arg \max_{\psi \in \Psi} \left\{ \mathcal{A}(\psi) + \alpha \frac{\mathcal{N}^{t-1}}{\mathcal{N}^t} \frac{\mathcal{P}(\Psi)}{\mathcal{P}(\psi)} \right\}, \quad (7)$$

where $\mathcal{A}(\cdot)$ is the top-1 accuracy of test dataset, \mathcal{N}^t denotes the number of data on growth step t , $\mathcal{P}(\cdot)$ represents the total amount of model parameters and α is the trade-off hyperparameter. When we get a large number of new coming data, $\frac{\mathcal{N}^{t-1}}{\mathcal{N}^t}$ will get small, which means that we focus more on the performance and encourage the growth of models. As our growth space is relatively small, traversal evaluation of all the candidate architectures is affordable. Therefore, we simply test all candidate subnets in the growth space Φ using the parameters inherited from the supernet Ψ and select the optimal one for the next stage via the composite metric of Equation 7.

Phase 3: Selected Model Training To alleviate the influence of supernet training on the grown model, we also use PIM to initialize the parameter ω of selected model ψ^* :

$$\omega = \beta \cdot \omega_{super}^* + \gamma \cdot \omega_{rand}. \quad (8)$$

After the parameter inheriting, we keep training the selected model, to obtain its final optimal parameters:

$$\omega^* = \arg \min_{\omega} \mathcal{L}(\psi^*, \omega; \mathcal{D}^t). \quad (9)$$

4. Experiments

In this section, we firstly describe the experiment setting and implementation details (Section 4.1). Then we show the results on zero-shot image classification (Section 4.2) and image-text retrieval (Section 4.3). Finally, we conduct ablation study on our GrowCLIP to validate the effectiveness of components (Section 4.4) and analyse (Section 4.5).

4.1. Experiment Setting

Benchmark Protocol Conceptual 12M (CC12M) [9] collects about 12 million image-text pairs from the Internet.

Growth step	Method	Training data	Parameter	Caltech101	CIFAR10	CIFAR100	DTD	Flowers102	Food101	OxfordPets	SUN397	ImageNet	Average
Step 1	CLIP (TWP)	3M	149.6M	42.8	37.6	14.5	10	9.3	16.1	21.6	22.0	14.0	20.9
	CLIP (TFS)[51]	3M	149.6M	42.8	37.6	14.5	10	9.3	16.1	21.6	22.0	14.0	20.9
	MS-CLIP (TFS)[65]	3M	129M	47.1	35.5	14.4	10.0	12.9	17.3	30.1	24.7	18.1	23.3
	CLIP (SAP)[4]	3M	149.6M	42.8	37.6	14.5	10	9.3	16.1	21.6	22.0	14.0	20.9
	NAS-S1	3M	30.0M	40.5	31.2	13.3	9.0	8.9	15.8	20.2	23.2	13.7	19.5
	GrowCLIP-S1 (ours)	3M	30.0M	40.5	31.2	13.3	9.0	8.9	15.8	20.2	23.2	13.7	19.5
Step 2	CLIP (TWP)	6M	149.6M	50.3	36.1	16.0	11.3	12.4	21.2	27.1	25.8	17.9	24.2
	CLIP (TFS)[51]	6M	149.6M	58.5	52.0	22.3	12.1	13.1	25.0	33.6	30.1	20.9	29.7
	MS-CLIP (TFS)[65]	6M	129M	60.6	50.0	18.3	13.4	17.4	27	42.4	32.6	25.2	31.8
	CLIP (SAP)[4]	6M	149.6M	61.9	49.0	21.4	12.4	15.5	24.6	35.4	31.6	22.6	30.5
	NAS-S2	6M	96.9M	59.4	41.8	18.3	13.3	15.8	26.7	37.3	33.8	24.4	30.1
	GrowCLIP-S2 (ours)	6M	116.6M	57.7	48.0	21.2	13.7	18.0	29.5	39.5	34.5	25.7	32.0
Step 3	CLIP (TWP)	9M	149.6M	55.2	40.1	19.7	13.1	15.7	23.8	30.9	28.9	20.4	27.5
	CLIP (TFS)[51]	9M	149.6M	63.3	54.6	24.5	14.5	17.5	29.5	36.9	33.8	25.3	33.3
	MS-CLIP (TFS)[65]	9M	129M	65.0	48.2	22.5	13.9	20.6	31.0	42.6	35.7	29.2	33.0
	CLIP (SAP)[4]	9M	149.6M	67.8	53.9	26.7	15.1	21.2	32.9	41.4	36.0	28.2	35.9
	NAS-S3	9M	287.4M	66.7	46.8	22.2	14.9	19.7	35.4	46.1	39.3	30.1	35.8
	GrowCLIP-S3 (ours)	9M	168.6M	69.6	53.9	25.8	18.4	18.7	36.3	46.2	41.6	32.6	38.1
Step 4	CLIP (TWP)	12M	149.6M	57.5	44.4	20.5	12.1	14.2	24.1	34.7	30.3	22.2	28.9
	CLIP (TFS)[51]	12M	149.6M	65.9	57.3	29.7	16.3	18.5	32.9	45.5	36.7	28.3	36.8
	MS-CLIP (TFS)[65]	12M	129M	65.3	50.9	26.3	16.2	21.9	35.9	50.1	38.4	32.0	37.4
	CLIP (SAP)[4]	12M	149.6M	69.8	60.6	31.9	12.4	21.0	36.7	47.0	40.4	32.2	39.1
	NAS-S4	12M	325.9M	70.8	53.9	28.3	16.7	21.7	39.5	51.2	41.3	34.2	39.7
	GrowCLIP-S4 (ours)	12M	188.4M	71.9	60.7	28.3	17.3	23.3	42.5	52.4	45.5	36.1	42.0

Table 3. Top-1 accuracy(%) of zero-shot image classification on 9 datasets. Our GrowCLIP outperforms the baseline methods in terms of average top-1 accuracy over 9 datasets at each growth step except step 1.

The CC12M is randomly divided into 4 subsets. We select one subset as the training set for the first growth step and adds one subset to the training set at each subsequent growth step such that there are total 4 growth steps in our experiments. At each growth step, we evaluate the corresponding model on the downstream tasks including zero-shot image classification and zero-shot image-text retrieval. The grown architecture of each growth step is shown in Appendix.

Comparison Methods To demonstrate the effectiveness of our method, we compare it with the following baselines: (i) Training with pre-training (TWP) is a usual method in transfer learning. (ii) Training from scratch (TFS) [51] is used as a basic method to train language-image pre-trained model. (iii) MS-CLIP [65] is a hand-crafted framework with a shareable module and modality-specific auxiliary module (the early specialization). (iv) The shrink and perturb (SAP) [4] is a method of parameter inheriting in warm-starting learning. (v) Neural architecture search (NAS) [48, 10] is an important research field of auto machine learning, in which we use the same search space as the growth space of GrowCLIP.

Implementation Details A dual-stream model follow-

ing CLIP [51] without shared encoder is used as basic architecture at the first growth step. The image encoder is stacked with 6 transformer blocks with the following hyperparameters: number of self-attention heads is 6 and the hidden layer size is 384. As for the text encoder, 6 transformer blocks are stacked and the number of the attention heads and the width are 4 and 256 respectively. The resolution of input in the images encoder is resized to 224 and the maximum sequence in the text encoder is limited to 77. The center crop is the only data augmentation method for training. We use the prompts like CLIP¹ on the zero-shot image classification. We adopt LAMB optimizer [66] with learning rate 0.01 and weight decay 0.0001, and cosine learning rate schedule [44] with a 4000-iteration linear warmup. For the hyperparameters, the proportion of the old model and the new one in the parameter inheriting $\beta = 0.3$, $\gamma = 0.001$. The model parameter coefficient in growth architecture selection $\alpha = 0.5$. We train the model with a total batch size of 1536 for 30 epochs at each growth step, which includes 2 epochs for supernet fine-tuning, 2 epochs for su-

¹<https://github.com/openai/CLIP/blob/main/data/prompts.md>

Growth step	Method	Flickr30K						MSCOCO					
		image-to-text			text-to-image			image-to-text			text-to-image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Step 1	CLIP (TWP)	19.6	45.3	56.4	14.3	33.5	44.1	10.7	26.3	36.8	7.0	19.1	27.5
	CLIP (TFS) [51]	19.6	45.3	56.4	14.3	33.5	44.1	10.7	26.3	36.8	7.0	19.1	27.5
	MS-CLIP (TFS)[65]	27.4	51.8	63.3	18.1	39.2	50.7	12.9	31.9	43.2	8.8	23.5	33.1
	CLIP (SAP) [4]	19.6	45.3	56.4	14.3	33.5	44.1	10.7	26.3	36.8	7.0	19.1	27.5
	NAS-S1	22.8	45.2	57.8	13.5	34.2	46.0	10.6	27.4	37.5	7.0	19.9	28.8
	GrowCLIP-S1 (ours)	22.8	45.2	57.8	13.5	34.2	46.0	10.6	27.4	37.5	7.0	19.9	28.8
Step 2	CLIP (TWP)	26.6	51.0	62.0	17.4	39.8	50.7	12.8	30.5	40.8	8.3	22.5	31.9
	CLIP (TFS) [51]	31.4	57.7	68.1	21.2	46.7	57.1	17.1	37.4	48.7	10.9	27.3	37.4
	MS-CLIP (TFS)[65]	35.8	65.1	75.0	25.8	53.1	64.6	19.3	43.0	55.2	13.1	31.5	42.6
	CLIP (SAP) [4]	33.6	60.9	72.0	24.1	48.2	59.3	17.1	38.0	50.2	10.9	28.0	28.1
	NAS-S2	36.9	63.0	74.0	25.2	51.7	62.3	18.4	41.6	53.7	12.9	30.9	41.7
	GrowCLIP-S2 (ours)	37.5	63.7	74.2	25.9	51.1	62.8	19.1	42.8	54.7	12.9	30.8	41.9
Step 3	CLIP (TWP)	28.3	54.4	66.6	20.6	42.2	54.7	14.4	34.1	45.0	9.6	25.2	34.9
	CLIP (TFS) [51]	37.5	63.5	74.2	26.0	52.1	64.0	19.7	43.2	55.1	13.3	31.8	42.6
	MS-CLIP (TFS)[65]	40.0	67.7	78.2	29.8	57.0	68.8	22.9	47.9	59.1	15.3	35.2	46.6
	CLIP (SAP) [4]	40.7	68.0	78.3	28.1	55.3	67.1	22.8	45.7	58.8	14.2	33.4	44.6
	NAS-S3	43.2	72.8	80.7	32.1	59.9	70.2	23.4	48.3	61.0	15.7	36.8	48.3
	GrowCLIP-S3 (ours)	44.6	73.1	80.7	33.2	60.7	70.9	23.5	48.8	61.0	15.8	36.9	48.4
Step 4	CLIP (TWP)	30.2	56.3	66.3	21.7	46.6	57.6	15.6	36.2	47.8	10.4	26.5	36.9
	CLIP (TFS) [51]	41.5	68.6	80.0	30.0	56.9	68.0	21.8	45.5	57.3	14.3	33.8	45.1
	MS-CLIP (TFS)[65]	44.3	72.8	81.1	32.6	61.3	72.1	25.2	50.5	62.0	16.6	37.7	49.0
	CLIP (SAP) [4]	46.9	74.7	83.7	33.7	61.3	72.1	25.2	50.8	62.8	16.6	37.4	49.1
	NAS-S4	49.6	75.2	83.3	37.0	65.2	75.4	26.8	53.0	64.6	17.7	39.6	51.5
	GrowCLIP-S4 (ours)	50.8	75.8	84.0	37.2	65.3	76.2	27.6	53.2	64.9	18.1	40.3	52.0

Table 4. Results of zero-shot image-text retrieval on Flickr30K and MSCOCO datasets. R@K means top-K recall.

pernet training and 26 epochs for grown model training. For the baselines, the CLIP adopts the same usual architecture (ViT-B/16), which has 12-layer 768-wide transformer blocks with 12 attention heads for image encoder and 12-layer 512-wide transformer blocks with 8 attention heads for text encoder and are trained for 30 epochs. For NAS, we first train the supernet for 30 epochs to select the best architecture, and then train the selected model from scratch for another 30 epochs. All experiments are performed on 16 NVIDIA Tesla V100 GPUs.

4.2. Zero-shot Image Classification

At each growth step, we evaluate the model on zero-shot image classification task. The downstream datasets includes 9 typical settings, Caltech101 [34], CIFAR10 [31], CIFAR100 [31], Describable Textures Dataset (DTD) [12], Flowers102 [46], Food101 [6], OxfordPets [47], SUN39 [5] and ImageNet [14]. We apply prompt ensemble evaluation like CLIP for each dataset. Results are given in Table 3. Our model performs growth process at each growth step except step 1, where the parameter grows from 30.0M to 188.4M. It outperforms the baseline methods in terms of average top-1 accuracy over 9 datasets at the last three growth steps. For example, at the third growth step, our model improves the average top-1 accuracy from 35.9% to 38.1% (+2.2%).

4.3. Zero-shot Image-Text Retrieval

The zero-shot image-text retrieval consists of two sub-tasks: image-to-text retrieval and text-to-image retrieval. We evaluate the model of each growth step on two retrieval benchmark datasets: Flickr30K [49] and MSCOCO [39]. As shown in Table 4, our model gets better as model grows and data grows. At the last two growth steps, it can be observed that our method has great performance on all metrics. Particularly, our method improves from 49.6% to 50.8% (+1.2%) for top-1 image-to-text recall on Flickr30K dataset and from 26.8% to 27.6% (+0.8%) on MSCOCO at the last growth step.

4.4. Ablation Study

Here, we conduct the ablation study on our GrowCLIP. All experiments are built at the growth step 2 and we evaluate the models on zero-shot image classification of ImageNet. To verify the effectiveness of each module in our proposed GrowCLIP, we conduct ablation experiments at some core steps mentioned in Section 3.4 as follows. (i) Parameter inheriting with momentum in phase 1 (PIM@1). (ii) Supernet fine-tuning in phase 1 (ST). (iii) Growth architecture selection in phase 2 (GAS). (iv) Parameter inheriting with momentum in phase 3 (PIM@3). The accuracy on zero-shot image classification of ImageNet is re-

PIM@1	ST	GAS	PIM@3	Para.	Acc.
✓	✓	✓	✓	116.6M	25.7
✓	✓	✓	✗	116.6M	24.5
✓	✓	✗	✗	109.5M	22.6
✓	✗	✗	✗	109.5M	21.1
✗	✗	✗	✗	109.5M	20.8

Table 5. Ablation analysis of each module in GrowCLIP at Growth Step 2. (Para.: parameter of the model. Acc.: top-1 accuracy (%) of zero-shot image classification on ImageNet.)

	Para.	Acc.
CLIP-ViT-B/16 (TFS)	149.6M	28.3
CLIP-ViT-L/16 (TFS)	431.1M	29.9
GrowCLIP-S4 (TFS)	188.4M	33.3
GrowCLIP-S4 (ours)	188.4M	36.1

Table 6. Ablation analysis of GrowCLIP-S4.

ported for comparison and the results of the ablation study are shown in Table 5. As can be seen, all above four components have positive effects on the final results. Specifically, the supernet training will influence the grown model without PIM@3, which results in 1.2% drop. Compared with GAS method, we randomly choose an architecture in the growth space, which has a worse performance from 24.5% to 22.6%. With ST, we observe a remarkable improvement of 1.5%. GrowCLIP with PIM@1 solves the problem of local minimum dilemma and brings a 0.3% improvement.

4.5. Analysis

Why can GrowCLIP improve the performance?

Compared with baseline models, the gains of our GrowCLIP come from two folds: (i) GrowCLIP can search the architecture that matches the current data with Growth architecture selection (GAS). (ii) Parameter inheriting with momentum (PIM) benefits GrowCLIP from yesterday’s model. To demonstrate their effectiveness, we train the architecture selected by GrowCLIP at step 4 from scratch, and the result is shown in Table 6. Compared with ViT-B/16, the growth architecture selection can bring 5% improvement on zero-shot image classification of ImageNet. Furthermore, it is even better than ViT-L/16, which is two and a half times as big as our model. More experiments have proved the effectiveness of GAS in the appendix. Comparing our GrowCLIP-S4 with GrowCLIP-S4 (TFS), we observe a remarkable improvement of 2.8% with PIM.

Why can GrowCLIP improve the training efficiency?

Our GrowCLIP can improve training efficiency, due to GAS and PIM, and the result is shown in Table 7. (i) Compared with the fixed architecture method, GrowCLIP has a smaller model with the GAS, so we reduce training costs

	Para.	Acc.	Epoch	Cost
CLIP-S2(TFS)	149.6M	20.9	30	336
GrowCLIP-S2 (ours)	116.6M	25.7	30	288
GrowCLIP-S4 (TFS)	188.4M	33.3	30	1230
GrowCLIP-S4 (ours)	188.4M	33.3	9	722
NAS-S4	325.9M	34.2	60	5280
GrowCLIP-S4 (ours)	188.4M	36.1	30	1583

Table 7. Efficiency analysis. (Cost: GPU hours)

	Para.	Acc.
GrowCLIP-S1	431.1M	14.6
GrowCLIP-S2	431.2M	26.2

Table 8. The architecture of GrowCLIP-S1 with bigger initialized model (ViT-L/16).

at the early steps (48 GPU hours saved at step 2). At the later steps, GrowCLIP uses fewer epochs to achieve good performance because of PIM. As shown in Figure 1, GrowCLIP has far outperformed CLIP (TFS) only after supernet training. To demonstrate the effectiveness of PIM, we compare GrowCLIP-S4 with GrowCLIP-S4 (TFS). GrowCLIP only uses 9 epochs to achieve comparable performance with PIM. (ii) Compared with the NAS method, GrowCLIP inherits the parameters of the old model with momentum, which only takes 2 epochs to fine-tune the supernet and 2 epochs to train the supernet. In contrast, one-shot NAS methods consume about 3.3X resources.

Can GrowCLIP find the relationship between model and data?

When the model scale is sufficient large, there is no need to enlarge the model. We conduct the experiment started from ViT-L/16. And the result in Table 8 shows that the larger model startpoint hardly grows at step 2, which means that our GrowCLIP can find the relationship between the model and data. In our online learning scenario, the initial architecture of the model is not important due to our growth pipeline. If the model scale is smaller at the beginning, GrowCLIP will select a larger enough model for growth. In contrast, if the model scale is too large initially, GrowCLIP will stop growing.

What is the influence when growth happens in different dimensions?

As shown in Table 9, different growth dimensions have different influences on the performance. By analyzing the experimental results, we conclude 4 laws, which can provide a reference for the design of cross-modal model architecture: (i) Enlarging the smaller model bring a greater improvement than the bigger one. For example, when the growth appears in transformer blocks in the image encoder, the accuracy can gain 0.93% improvement at step 2 but only 0.12% at step 4. (ii) We need to leave the image

Dimension	Step 2		Step 3		Step 4	
	Para.	Acc.	Para.	Acc.	Para.	Acc.
+ 4 b^I	+7.1M	+0.93	+19.7M	+0.46	+19.7M	+0.12
+ 4 h^I	+20.1M	+0.34	+67.3M	+0.12	+105.1M	+0.03
+ 2 l^I	+0.1M	+0.84	+0.1M	+0.07	+0.1M	+0.10
+ 4 b^T	+3.2M	+0.01	+12.6M	+0.08	+12.6M	+0.01
+ 4 h^T	+27.0M	+0.22	+45.0M	+0.01	+69.5M	-0.04
+ 4 b^S	+7.1M	+1.03	+19.7M	+0.19	+19.7M	-0.07

Table 9. The ablation of growth space corresponding to different dimensions. (+ 4 b^I : the number of transformer blocks in image encoder is increased by 4)

encoder with more capacity. Compared with text encoder, the improvement of growing the image encoder is more significant at each step. (iii) The convolutional layers is a very important component in the image encoder, which can help the model to better extract visual feature. (iv) The shared encoder can enhance the degree of cross-modal fusion and bring a remarkable improvement. However, do not use a large number of transformers in the shared encoder, or the model become redundant with no noticeable performance improvement.

5. Conclusion

In this paper, we study the online learning case in cross-modal pre-trained models. The small-scale architecture may limit the final performance due to the limited parameters when data grows. Based on this real-world scenario, we propose a data-driven automatic model growing algorithm for language-image pre-training - GrowCLIP, which upgrades the model size when abundant new data arrives. We adopt parameter inheriting with momentum (PIM) to utilize the knowledge in yesterday’s model and seek out the optimal growth architecture at each growth step with growth architecture selection (GAS). GrowCLIP improves the performance on downstream tasks, including zero-shot image classification and retrieval. The effectiveness of each component of GrowCLIP is proved on ablation studies. And we provide a reference for the design of cross-modal model architecture.

Limitations In this work, we only prove the effectiveness of GrowCLIP with public pre-training dataset - CC12M. In future works, we will extend our method to a real-world scenario, where the VLP model keeps training with constantly updated data crawled from the web.

Acknowledgment

This work was supported in part by National Key R&D Program of China under Grant No.2020AAA0109700, Guangdong Outstanding Youth Fund (Grant No.

2021B1515020061), Shenzhen Science and Technology Program (Grant No.RCYX20200714114642083), Shenzhen Fundamental Research Program(Grant No. JCYJ20190807154211365), Nansha Key RD Program under Grant No.2022ZD014 and Sun Yat-sen University under Grant No.221gqb38 and 76160-12220011. We thank MindSpore for the partial support of this work, which is a new deep learning computing framework².

²<https://www.mindspore.cn/>

References

- [1] J. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*.
- [2] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European conference on computer vision*, 2018.
- [3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Zitnick, and D. Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision*, 2015.
- [4] J. Ash and R. Adams. On warm-starting neural network training. 2020.
- [5] A. Barriuso and A. Torralba. Notes on image annotation. Preprint arXiv:1210.3448, 2012.
- [6] L. Bossard, M. Guillaumin, and L. Gool. Food-101—mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.
- [7] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020.
- [8] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2019.
- [9] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Computer Vision and Pattern Recognition*, 2021.
- [10] M. Chen, H. Peng, J. Fu, and H. Ling. Autoformer: Searching transformers for visual recognition. In *International Conference on Computer Vision*, 2021.
- [11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.
- [12] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition*, 2014.
- [13] M. Cornia, L. Baraldi, and R. Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *Computer Vision and Pattern Recognition*, 2019.
- [14] J. Deng. A large-scale hierarchical image database. 2009.
- [15] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.
- [16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [17] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition*, 2006.
- [18] J. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. 2020.
- [19] Z. Guo, B. Avila Pires, B. Piot, J. Grill, F. Altché, R. Munos, and M. Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, 2020.
- [20] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, 2020.
- [21] R. Hadsell, D. Rao, A. Rusu, and R. Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 2020.
- [22] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2016.
- [23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition*, 2022.
- [24] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition*, 2020.
- [25] X. Hu, Z. Gan, J. Wang, Z. Yang, Z. Liu, Y. Lu, and L. Wang. Scaling up vision-language pre-training for image captioning. Preprint arXiv:2111.12233, 2021.
- [26] X. Hu, X. Yin, K. Lin, L. Wang, L. Zhang, J. Gao, and Z. Liu. Vivo: Surpassing human performance in novel object captioning with visual vocabulary pre-training. 2020.
- [27] C. Jia, Y. Yang, Y. Xia, Y. Chen, Z. Parekh, H. Pham, Q. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.
- [28] J. Kaplan, S. McCandlish, T. Henighan, T. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. Preprint arXiv:2001.08361, 2020.
- [29] W. Kim, B. Son, and I. Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, 2021.
- [30] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, 2017.
- [31] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [32] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*, 2020.
- [33] C. Li, B. Zhuang, G. Wang, X. Liang, X. Chang, and Y. Yang. Automated progressive learning for efficient training of vision transformers. In *Computer Vision and Pattern Recognition*, 2022.
- [34] F. Li, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 2006.

- [35] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.
- [36] L. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang. Visualbert: A simple and performant baseline for vision and language. Preprint arXiv:1908.03557, 2019.
- [37] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 2020.
- [38] Y. Li, F. Liang, L. Zhao, Y. Cui, W. Ouyang, J. Shao, F. Yu, and J. Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. Preprint arXiv:2110.05208, 2021.
- [39] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.
- [40] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L. Li, F. Li, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *European conference on computer vision*, 2018.
- [41] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. Preprint arXiv:1806.09055, 2018.
- [42] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu. Cptr: Full transformer network for image captioning. Preprint arXiv:2101.10804, 2021.
- [43] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Preprint arXiv:1907.11692, 2019.
- [44] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. 2016.
- [45] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee. 12-in-1: Multi-task vision and language representation learning. In *Computer Vision and Pattern Recognition*, 2020.
- [46] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing*, 2008.
- [47] O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition*, 2012.
- [48] H. Peng, H. Du, H. Yu, Q. Li, J. Liao, and J. Fu. Cream of the crop: Distilling prioritized paths for one-shot neural architecture search. In *Advances in Neural Information Processing Systems*, 2020.
- [49] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *International Conference on Computer Vision*, 2015.
- [50] R. Pogodin and T. Lattimore. On first-order bounds, variance and gap-dependent bounds for adversarial bandits. In *Uncertainty in Artificial Intelligence*, 2020.
- [51] A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.
- [52] E. Real, A. Aggarwal, Y. Huang, and Q. Le. Regularized Evolution for Image Classifier Architecture Search. In *AAAI Conference on Artificial Intelligence*, 2019.
- [53] E. Real, S. Moore, A. Selle, S. Saxena, Y. Suematsu, J. Tan, Q. Le, and A. Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning*, 2017.
- [54] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *International Conference on Computer Vision workshops*, 2009.
- [55] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 2012.
- [56] S. Song and B. Choi. A study on continuous follow-the-leader (ftl) gaits: an effective walking algorithm over rough terrain. *Mathematical biosciences*, 1989.
- [57] T. Srinivasan, T. Chang, L. Alva, G. Chochlakis, M. Rostami, and J. Thomason. Climb: A continual learning benchmark for vision-and-language tasks. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [58] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Computer Vision and Pattern Recognition*, 2019.
- [59] B. Thomee, D. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016.
- [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Neural Information Processing Systems*, 2017.
- [61] Haonan Wang, Minbin Huang, Runhui Huang, Lanqing Hong, Hang Xu, Tianyang Hu, Xiaodan Liang, and Zhenguo Li. Boosting visual-language models by exploiting hard samples. *arXiv preprint arXiv:2305.05208*, 2023.
- [62] S. Xie, H. Zheng, C. Liu, and L. Lin. SNAS: Stochastic Neural Architecture Search. In *International Conference on Learning Representations*, 2019.
- [63] Shipeng Yan, Lanqing Hong, Hang Xu, Jianhua Han, Tinne Tuytelaars, Zhenguo Li, and Xuming He. Generative negative text replay for continual vision-language pretraining. In *European Conference on Computer Vision*, pages 22–38. Springer, 2022.
- [64] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu. Filip: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022.
- [65] Haoxuan You, Luwei Zhou, Bin Xiao, Noel Codella, Yu Cheng, Ruochen Xu, Shih-Fu Chang, and Lu Yuan. Learning visual representation from modality-shared contrastive language-image pre-training. In *European Conference on Computer Vision*, pages 69–87. Springer, 2022.

- [66] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020.
- [67] F. Yu, J. Tang, W. Yin, Y. Sun, H. Tian, H. Wu, and H. Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3208–3216, 2021.
- [68] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. Preprint arXiv:2205.01917, 2022.
- [69] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.
- [70] B. Zoph, V. Vasudevan, J. Shlens, and Q. Le. Learning Transferable Architectures for Scalable Image Recognition. In *Computer Vision and Pattern Recognition*, 2018.